Improving Information Retrieval in Multiwriter Scenario by Exploiting the Similarity Graph of Document Terms

Pau Riba, Anjan Dutta, Sounak Dey, Josep Lladós and Alicia Fornés
Computer Vision Center, Computer Science Department,
Universitat Autònoma de Barcelona, Barcelona, Spain
Email: {priba, adutta, sdey, josep, afornes}@cvc.uab.es

***Abstract*—Information Retrieval (IR) is the activity of obtaining information resources relevant to a questioned information. It usually retrieves a set of objects ranked according to the relevancy to the needed fact. In document analysis, information retrieval receives a lot of attention in terms of symbol and word spotting. However, through decades the community mostly focused either on printed or on single writer scenario, where the state-of-the-art results have achieved reasonable performance on the available datasets. Nevertheless, the existing algorithms do not perform accordingly on multiwriter scenario. A graph representing relations between a set of objects is a structure where each node delineates an individual element and the similarity between them is represented as a weight on the connecting edge. In this paper, we explore different analytics of graphs constructed from words or graphical symbols, such as diffusion, shortest path, etc. to improve the performance of information retrieval methods in multiwriter scenario.**

***Index Terms*—document terms, information retrieval, affinity graph, graph of document terms, multiwriter, graph diffusion**

I. Introduction

The large amount of document repositories and the emergence of new services for document analytics require solutions for efficient search and indexation by content. Information spotting methods have experienced a qualitative increase as drivers of document information retrieval. Digital mailroom solutions, search in historical document archives or sketching interfaces for graphical documents are some examples of use cases. Retrieval by content generally assumes that documents are segmented in *terms* (words, symbols, meaningful units, patches) and given a *query term*, the precision of the result depends on the location of the retrieved instances in the corresponding pages.

In multiwriter scenarios, the inherent variability among instances of the same class results in a low precision. To tackle this issue, some strategies are addressed to organize the terms in the database according to their similarity. The simplest organization is a clustering of terms, which is already applied to many retrieval scenarios. As the early word spotting methods proposed by Rath and Manmatha [13] when the database is organized in clusters, given a query, the retrieved terms are extracted from clusters that have a representative close to the query. Another typical strategy is *query expansion*. It is based on expanding the query by adding additional queries using techniques such as relevance feedback or reformulating the retrieval using the closest terms in the database. Almazán et al. [1] proposed a query expansion scheme for word spotting.

In information retrieval, different approaches have been proposed to improve the ranking of retrieved objects based on the use of similar objects existing in the database [18]. Similar database objects define a data manifold structure, thus these objects form a context through which the search can be organized. The data is represented as a graph with edge weights determined by pairwise similarities of the objects. The motivation of our work is the improvement of retrieval rankings in an information spotting context by organizing the terms of the database in a similarity graph structure. Thus, an attributed affinity graph is constructed where nodes are the terms of the database documents (assuming that they have been segmented), and edges correspond to their affinity or similarity. Node attributes are extracted from the properties of terms (shape and local topology) and edge attributes correspond to the similarity measure. In a different scenario, Puigcerver *et al.* [12] constructed a word graph with the different decoding possibilities in a line-wise recognition scheme with HMMs. A word spotting strategy was proposed based in the traversal of this word graph.

The main contribution of this work is the proposal of graph analytics, *viz.*, *shortest path* on graph[1] and *graph diffusion* on affinity graph to improve the retrieval ranking of terms in multiwriter databases. Particularly, the similarity between the query and each database term is reevaluated in the existing context. The *context* of a term signifies the set of other terms most similar to it, which usually includes sufficient variability to model distortion. Graph diffusion propagates the affinity information following the structure of the context. Intuitively, in a multiwriter scenario, a context of a term can be seen as the manifold consisting of terms of the same class written by different writers. The propagation of similarity follows the geodesic paths through the class. Additionally, we propose a retrieval approach based on double query. In some scenarios where two queries of different writers are available, the combination of the corresponding graph diffusion can be seen as expanding the queries through the graph path that connects one to the other passing through the instances of the different writers.

Figure 1 shows the spotting approach based on double query. Given a multiwriter document database, the proposed

[1]In this paper, unless otherwise specified, by *graph* we mean a basic graph whose edge set specifies the neighborhood and distance among the nodes. Differently, we specifically mention *affinity graph* where the adjacency matrix contains pairwise similarity values between objects.

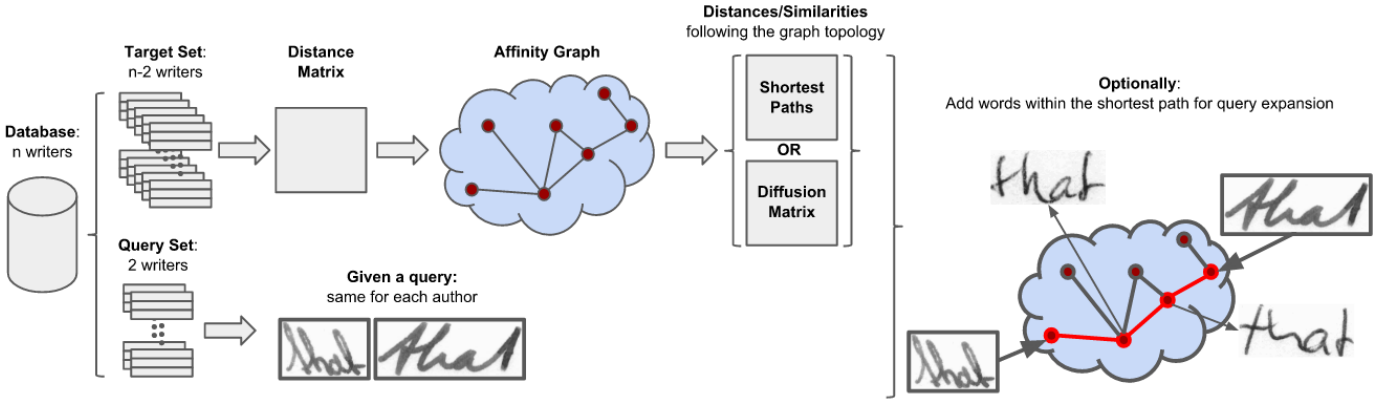

Fig. 1. Overview of our framework in the case of double query. Given an image database, the proposed methodology constructs an affinity graph to recompute a new distance matrix. Receiving two new authors and their corresponding queries, they are searched within the graph using the their nearest neighbor. Optionally, the shortest path can be used for query expansion.

methodology constructs an affinity graph to recompute a new distance matrix. Given two term queries corresponding to the same class, they are searched within the graph using their nearest neighbor. Optionally, the terms belonging to the shortest path can be used for query expansion.

We thoroughly evaluate the proposed affinity graph analytics for improving information retrieval in different multiwriter databases and scenarios. In particular, databases containing words and graphic symbols generated by multiple users have been considered for the method assessment.

The rest of this paper is organised as follows: Section II is devoted to introduce the concept of graph of terms or objects following the idea of affinity graph. Section III explains the image affinities computations within the graph. In Section IV, we explain the retrieval strategies applied to the constructed graph. Afterwards, Section V presents the validation and results of the proposed approach. Finally, Section VI draws the conclusion of the present work and future research related to the present topic is defined.

II. Graph Construction

An *attributed graph* is a $4-$tuple $G(V, E, L_V, L_E)$ comprising a set V of *vertices* together with a set $E \subseteq V \times V$ of *edges* and two *mappings* $L_V : V \rightarrow \mathbb{R}^m$ and $L_E : E \rightarrow \mathbb{R}^n$ which respectively assign attributes to the nodes and edges. Hence, representing a real data with graphs needs defining the set of nodes and edges. Numerous studies have been done in the recent years regarding the reliability of graph construction [16]. There are essentially two categories of approaches depending on whether the graph construction is unsupervised or semi-supervised/supervised. The unsupervised methods divide themselves between static and adaptive techniques, where in static graphs the connected nodes are selected using hard decision and the pairwise similarities or affinities are computed without consideration of other data points. In this paper, we construct the static graphs to represent the data manifold created by the document terms, where each node delineates a particular term and each edge codifies a (dis)similarity measure.

A. Attribute Functions

Representing document terms with graphs requires the definition of the node and edge attribute functions, based in statistical analysis of the underlying data. We propose to label each node using an image descriptor encoding the corresponding term with an n-dimensional feature vector. Thus, the attributes of edges joining two nodes depend on the distance computation. The following *four* image descriptors have been used in the evaluation scenarios:

Blurred Shape Model (BSM) [7] is a shape descriptor which can be considered as a weighed zoning descriptor. The image is divided in a grid of n x m equal-sized subregions. Each bin receives votes from the foreground pixels in the image region, and also from the neighboring ones. This contribution is weighted according to the distance to the center of each bin. In other words, each bin encodes the probability of pixel densities.

Local binary patterns (LBP) is a texture descriptor which takes advantage of pooling of the original LBP in the adaptive regions [5]. These regions are determined by a pyramidal grid which is recursively updated through the calculation of image geometric centroids to calculate the feature vector for matching.

Scale Invariant Feature Transform (SIFT) [10] is a robust image descriptor that extracts a set of visual signatures from a given image. It can be computed on the detected regions as well as on regular grid. In this work, we use SIFT that is computed on regular grid filtered by high gradient response. The extracted descriptors are then encoded by Fisher Vector to encode higher order statistics like a bag of visual words.

Pyramidal Histogram of Characters (PHOC) [2] is the notion of word embeddings, creating a joint embedding space for word images and representations of word strings. This is achieved by a combination of label embedding, attributes learning, and a common vectorial subspace regression. The

test strings corresponding to the word images are projected to $d-$dimensional binary space which encodes if a particular character appears in particular spatial region of the string. This source of character attributes projects the word images to $d-$dimensional space which is learnt using the powerful SIFT+FV representation as explained above, where each dimension encodes the probability of that word image containing a particular character. This descriptor is used for the use cases on word spotting.

B. Edge Construction

The set of edges E in G defines the adjacency or neighbourhood of the nodes in V. The nodes that are in a certain vicinity are typically connected by edges. Image descriptors as defined in Section II-A, provide the notion of distance and neighbourhood of visual objects. In this paper, the *Euclidean* or L^2*-distance* is only used for BSM descriptors and *cosine distance* is used for rest of the cases, which are respectively defined as follows:

$$d_E(a,b) = \sqrt{\|a\|^2 + \|b\|^2 - 2a.b} \tag{1}$$

$$d_C(a,b) = 1 - \frac{a \cdot b}{\|a\|\|b\|} \tag{2}$$

Given a distance matrix that contains pairwise distances between all pairs of node descriptors, there are several methodologies to construct a graph from that, some of them are described as below.

ε-threshold Graph Given a threshold ε, the corresponding graph of objects can be created connecting only those nodes whose distance is smaller than ε (i.e. $D < \varepsilon$). The obtained graph is symmetric, however, there is no control on the number of connections each node has.

k-NN Graph Given a $k \in \mathbb{N}$, each node is connected with its k nearest neighbours. This approach leads to a directed graph, however, it can be converted to an undirected one, if an edge exist in any direction. Therefore, every node will be connected at least with k nodes but can have more connections as well.

In this work, the graphs are constructed using the k-NN methodology in order to obtain graphs which are more stable in terms of connections per node. In particular, we perform experiments with $k = 4, 8$ and 16 (see section V).

C. Affinity Graph

The affinity between two objects indicates a measure of similarity between them. Therefore, the graph having pairwise similarities between all the nodes is called *affinity graph*. The nodes of a static graph representing an object commonly delineate the object parts. Here the node and edge attributes are usually the outcome of the statistical analysis of the object parts. Although similarity or affinity between two objects can be defined in many ways, usually in some sense, they are the suitable inverse of corresponding distance metrics. For example, a very basic similarity can be simply the negative of the Euclidean distance metric:

$$s(x,y) = -\|x - y\|_2$$

Similarly, cosine similarity between two objects is defined as the subtraction of cosine distance (Eqn. (2)) from 1:

$$s_C(a,b) = \frac{a \cdot b}{\|a\|\|b\|} = 1 - d_C(a,b)$$

Likewise, the affinities can be computed under any common kernel functions, such as, radial basis function (RBF), histogram intersection etc. In this paper, the affinities between two nodes v_i and v_j are computed with radial basis function as follows:

$$A(v_i, v_j) = e^{-\frac{d^2}{2\sigma}} \tag{3}$$

where d is either cosine or Euclidean distance. The parameter σ that controls the node neighborhood is set to 1.

III. Graph Analytics

The constructed graph of pairwise distances or affinities between terms can be seen as a term network. Thus, the navigation through this network and the extraction of some properties allows to improve the retrieval ranking performance. This Section proposes to recompute the initial distance matrix applying two distinct methodologies on both distance or affinity graph.

A. Shortest Path

A shortest path between two nodes v_i and v_j in a graph G is the path that visits the minimum number of nodes between v_i and v_j. In terms of weights on edges, shortest path between two nodes in a graph can be defined as the path that minimizes the sum of weights of the constituent edges. Shortest path has been applied to many applications, such as, finding the direction between two physical locations or finding an optimal sequence of choices to reach a certain goal state in a non-deterministic abstract machine. Realizing the importance of shortest path problem, many algorithms are proposed, among which algorithms proposed by Dijkstra [6], Fredman and Tarjan [8], FloydWarshall, Breadth-first search are some famous to name. Shortest path has also received importance for graph-based pattern recognition task, for example, the shortest path kernels define the similarity between two graphs in terms of shortest paths [3]. Furthermore, the betweenness centrality of a node, which measures how many shortest paths between other nodes pass through that node, is a graph invariant used for network analysis tasks.

B. Graph Diffusion

Given a graph G with V representing a set of objects and E as their affinities, the information fetched in graph diffusion contains geometric information about the data in V. Let us assume that each edge in E is labelled with the strength of the connection as $E(i,j) = A(v_i, v_j)$, where A is the function

for computing node affinities as in Eqn. (3). It is evident from the definition that A is symmetric and positivity preserving.

Diffusion on a graph can be thought of as a reversible Markov chain process on V [4] and can be interpreted as follows. The degree of each node in G can be computed as

$$D(v_i) = \sum_{j=1}^{n} A(v_i, v_j)$$

and the transition probability is defined as

$$P(v_i, v_j) = \frac{A(v_i, v_j)}{D(v_i)} \tag{4}$$

It is obvious that the transition matrix P in Eqn. (4) inherits the positivity preserving property, but not symmetric. However, we have gained a conservation property as below:

$$\sum_{j=1}^{n} P(v_i, v_j) = 1 \tag{5}$$

In the field of data analysis, the property of the transition matrix P has been studied thoroughly and revealed that P contains the geometric information about the data manifold in V [18]. This is because of the transitions that the matrix P defines directly reflect the local geometry defined by the immediate neighbours of each node in the graph of the data. For example, $P(v_i, v_j)$ represents the probability of transition in one time step from node v_i to node v_j and it is proportional to the edge weight $E(i, j)$. For any $t \geq 0$, the probability of transition from v_i to v_j in t time steps is given by $P^t(v_i, v_j)$, which is the t^{th} power of P. One of the main principles of graph diffusion is that the chain running forward in time, or equivalently, taking larger powers of P, allows to integrate the local geometry. Therefore it reveals relevant geometric structures of V at different scales. This is controlled by the scale parameter t.

Having established the fact that diffusion on a graph representing certain dataset of terms captures the geometry of the underlying manifold, diffusion on the tensor product graph (TPG) $\mathbb{G} = (V \times V, \mathbb{E})$ of the graph G reveals intrinsic relation between objects. This is justified by the fact that the edge weights of TPG relates four tuples of original nodes and hence contains high order information than the input graph. However, it has been shown that the diffusion on TPG of graph G can learn new affinities on the original graph [18], which can be computed as

$$E^* = \text{vec}^{-1}((I - \mathbb{E})^{-1}\text{vec}(I))$$

where I is an identity matrix and vec is an operator that stacks the columns of the input matrix one after the next one. From definition, it is evident that vec is a bijection and hence, its inverse exist. For avoiding the high space and time complexity of computing the diffusion on TPG, an iterative way is shown in [18] and that can be done as follows

$$Q^{(t+1)} = EQ^{(t)}E^T + I \tag{6}$$

where $Q^{(1)} = E$. Also, it has been proved that Eqn. (6) converges and produces the new affinities as

$$\lim_{t\to\infty} Q^{(t)} = Q^* = E^* = \text{vec}^{-1}((I - \mathbb{E})^{-1}\text{vec}(I))$$

In the context of our work, diffusion on TPG generates more reliable affinity values between any two words or graphic symbols. Intuitively, this procedure should bring the intra class words or symbols created by multiple writers in a close vicinity and push the inter class patterns far away.

IV. Retrieval Strategies

Two retrieval strategies are proposed, *viz.* single and double query. Having only one query, it comes from unseen author in the constructed graph. The proposed baseline, directly computes the distance from its descriptor to the whole target set. Using graph based techniques, the proposed framework finds the nearest node of the graphs in terms of feature distances, afterwards the distance to the rest of the queries is computed within the graph using either the shortest path or the diffusion. In the case of double query, same class but different author is used. The distance to the target set is computed separately for both queries using the same procedure as single query. Afterwards, the distances are combined through a minimum function, prioritising to be near to one of the queries. Optionally, using double query allows the computation of the shortest path whose nodes are expected to belong to the same class as the queries. Therefore, we propose to add those nodes to the set of queries using a similar procedure as query expansion. Figure 2 shows two examples of shortest paths between two queries. Path-wise instances are the ones used for query expansion. Notice that in some cases, wrong instances are found.

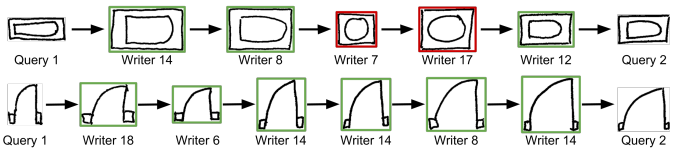

Fig. 2. Shortest path examples between two queries. Correct instances are marked in green whereas wrong examples are shown in red.

V. Experimental Validation

The proposed approach has been studied as an improved strategy for information retrieval in document analysis. Retrieval of symbols and words in documents written by multiple writers is studied to show the robustness of the proposed approach. The performance is evaluated using the mean average precision (mAP) metric.

A. Datasets

Three different multiwriter datasets have been used to evaluate the performance: a handwritten word dataset, and two multiwriter graphical symbols datasets.

The Architectural Symbol database [14] is a collection of hand drawn graphical symbols with 50 different classes.

Actually this dataset is generated by taking the symbols from the GREC dataset [9] and drawing them by different authors. This corpus is composed of online and offline instances drawn by a total of 22 users. Each user has drawn a total of 25 symbols and over 11 instances per symbol, which resulted in a total $5,000$ instances.

NicIcon [17] dataset is composed of $26,163$ handwritten symbols of 14 classes from 34 different writers. We have used the provided off-line data. The dataset is divided in three subsets (training, validation and test) for two different settings: writer dependent and writer independent. We have selected the writer independent set forcing that the writers from the queries are not in the target set.

IAM Handwriting Database [11] contains handwritten English text images provided at form, sentence and word levels. It is composed of 1,539 pages of scanned text which contains 5,685 isolated sentences, 13,353 text lines and 115,320 isolated words. In this work, 20 authors and 15 transcriptions have been used.

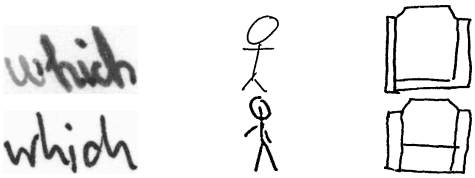

Fig. 3. Instances from 2 authors of the three used datasets. From left to right, IAM, Nicicon and Architectural symbol.

B. Information Retrieval

Information retrieval (symbols and words in document domain) is the task of retrieving the instances of a given query information. Usually the query and the images are represented by features invariant to visual distortions. Most retrieval techniques use statistical representations (e.g. HOG, SIFT) of the word images [2], [15]. However, there are also few approaches using structural representations.

Symbol retrieval: Firstly, let us focus on symbol databases using the BSM descriptor for isolated symbols. Performing their evaluation without separating all the instances of a same writer (only the query), gets a mean average precision of 68.43% and 50.24% respectively for architectural symbols and NicIcon datasets. Table I shows the performance achieved by both datasets using one and two queries. For the architectural symbols dataset, we have achieved an improvement with respect to the baseline of 25% and 18% for one and two queries, whereas in the NicIcon dataset, the improvement is 38% and 31%.

Regarding the graph size in terms of edges, the table shows that the best value is achieved with $k = 4$. Probably, the descriptor is not robust enough for a multiwriter scenario, therefore, choosing a too big k, produces noise that ends up with worse results. Once k is fixed, the best improvement is obtained using the explained diffusion technique. Since we have lots of examples for each symbol written by different authors, our target graph has densely connected regions that helps the diffusion algorithm.

Word retrieval: The variability among multiple writer instances of words is higher than in graphical symbols. For this experiment, more complex descriptors have been used, i.e. LBP, SIFT with Fisher Vectors and its projection to the PHOC space. The improvement achieved in this case is clearly small in comparison to the symbol retrieval experiment. The main issue is to deal with the lack of examples in the target set in order to construct the affinity graph. Some word classes contain a very few number of instances. This is because some authors have not written some words and in other cases there are few samples of each word.

Table I shows the performance using 3 different descriptors. In this case, the improvements with single and double query are respectively 0.2% and 1.84% for LBP, 7.56% and 1.66% for SIFT+FV and finally, 5.65% and 0.71% for PHOC. Therefore, we have improved the performance with all the descriptors. Notice that the best graph construction using k-NN is $k = 16$ with very robust descriptors, otherwise, the best performance is achieved with $k = 4$. This can be caused because a weaker descriptor introduces noise creating new connections. Regarding the node affinity computation, in general, we have obtained our best results using the diffusion algorithm. Except for word retrieval using LBP descriptor, introducing the shortest path between two given queries for query expansion leads to noise getting bad results. Figure 4 shows a summary of Table I showing a comparison between the baseline and the best results obtained (in bold in the Table) on each dataset both with single and double query strategies.

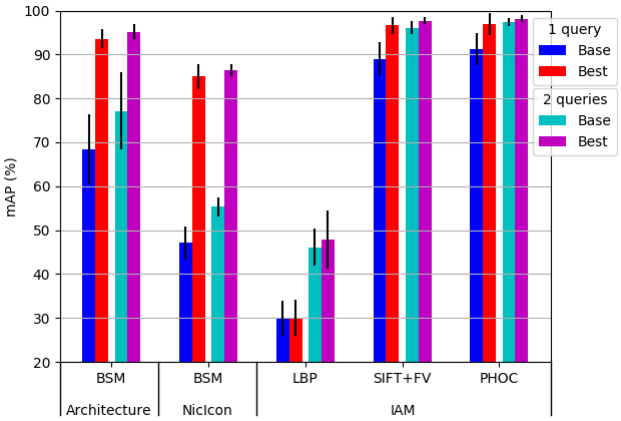

Fig. 4. Summary of Table I comparing the baseline and the best case for both, single and double query.

VI. Conclusions

In this paper, we have proposed to improve information retrieval in documents by means of graph based techniques. Our approach is evaluated in a multiwriter scenario for symbol and word retrieval. We propose to take advantage of the graph structure including a double query framework where the shortest path between queries can be optionally used for

TABLE I

MEAN AVERAGE PRECISION AND STANDARD DEVIATION IN % FOR THE PROPOSED FRAMEWORKS AND DATASETS. 1Q AND 2Q STANDS FOR THE TWO MODALITIES, USING JUST ONE QUERY OR TWO. SP AND DIFF MEANS THAT THE WORD SIMILARITY IS COMPUTED FOLLOWING THE SHORTEST PATH AND DIFFUSION TECHNIQUE RESPECTIVELY. FINALLY, QE ADDS THE SHORTEST PATH AS QUERIES FOR QUERY EXPANSION.

		Architectural Symbols			**NicIcon**			**IAM**								
Descriptor:		**BSM**			**BSM**			**LBP**			**SIFT + FV**			**PHOC**		
Graph:		**4-NN**	**8-NN**	**16-NN**	**4-NN**	**8-NN**	**16-NN**	**4-NN**	**8-NN**	**16-NN**	**4-NN**	**8-NN**	**16-NN**	**4-NN**	**8-NN**	**16-NN**
	Baseline	68.37±7.92			47.06±3.78			29.92±4.06			89.05±3.88			91.28±3.62		
1q	**SP**	89.39 ±3.23	88.07 ±3.89	85.32 ±5.00	71.18 ±2.43	69.38 ±2.69	67.06 ±2.96	**30.12 ±4.11**	29.79 ±3.90	29.88 ±3.92	93.41 ±2.39	94.08 ±2.46	93.95 ±2.44	90.88 ±3.37	93.72 ±2.94	95.27 ±2.67
	Diff	**93.61 ± 2.24**	93.31 ±2.26	91.24 ±2.95	**84.98 ±2.81**	84.32 ±2.95	82.96 ±2.92	28.62 ±3.78	28.25 ±3.71	27.32 ±3.31	95.45 ±2.03	96.57 ±2.08	**96.61 ±2.03**	92.58 ±3.18	95.69 ±2.91	**96.93 ±2.41**
	Baseline	77.16±8.81			55.35±2.16			46.08±4.18			96.07±1.45			97.46±0.92		
2q	**SP**	90.93 ±3.71	90.24 ±4.26	88.38 ±5.23	73.81 ±1.41	72.65 ±1.59	70.78 ±1.71	46.85 ±5.18	45.66 ±4.67	45.12 ±4.33	96.89 ±1.09	97.33 ±0.97	97.42 ±0.90	96.04 ±1.26	97.21 ±0.90	97.88 ±0.85
	Diff	**95.21 ±1.73**	95.11 ±1.85	93.42 ±2.70	**86.38 ±1.40**	85.88 ±1.40	84.88 ±1.41	44.36 ±4.40	43.90 ±4.01	42.58 ±3.90	96.74 ±1.12	97.59 ±1.02	**97.73 ±0.84**	95.51 ±1.28	97.54 ±0.88	**98.17 ±0.77**
	SP+QE	90.80 ±3.73	90.34 ±4.15	88.69 ±5.00	74.51 ±1.36	73.50 ±1.44	71.72 ±1.63	**47.92 ±6.63**	45.60 ±5.34	45.85 ±5.33	96.83 ±1.02	97.32 ±0.93	97.45 ±0.83	95.61 ±1.61	97.02 ±1.12	97.84 ±0.99
	Diff+QE	94.69 ±2.00	94.96 ±1.91	93.45 ±2.68	85.97 ±1.42	85.70 ±1.44	84.90 ±1.45	46.05 ±5.89	44.77 ±4.77	43.79 ±4.84	96.61 ±1.26	97.53 ±1.08	97.70 ±0.89	95.23 ±1.66	97.34 ±1.11	98.12 ±0.84

query expansion. It has been proved that creating an affinity graph can lead to a new distance matrix which improves the performance of the framework for both single and double query. The experimental results show that using graph diffusion techniques is a robust mechanism to compute node affinities which takes into account all the connections and not only the shortest path distance that can be easily affected by errors and noise.

Future work will be focused on improving the performance when there are few examples of each symbol for each author. Although it has been observed to be a bad scenario, we have been able to improve the baseline performance.

ACKNOWLEDGMENTS

This work has been partially supported by the European Union's research and innovation program under the Marie Skłodowska-Curie grant agreement No. 665919, the Spanish project TIN2015-70924-C2-2-R, a FPU fellowship FPU15/06264 from the Spanish Ministerio de Educación, Cultura y Deporte, the Ramon y Cajal Fellowship RYC-2014-16831 and the CERCA Programme/Generalitat de Catalunya.

REFERENCES

[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Segmentation-free word spotting with exemplar SVMs," *PR*, vol. 47, no. 12, pp. 3967 – 3978, 2014.

[2] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE TPAMI*, vol. 36, no. 12, pp. 2552–2566, 2014.

[3] K. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *ICDM*, 2005, pp. 74–81.

[4] R. R. Coifman and S. Lafon, "Diffusion maps," *ACHA*, vol. 21, no. 1, pp. 5–30, 2006.

[5] S. Dey, A. Nicolaou, J. Lladós, and U. Pal, "Local binary pattern for word spotting in handwritten historical document," in *S+SSPR*, 2016, pp. 574–583.

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[7] S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, and J. Lladós, "Blurred Shape Model for binary and grey-level symbol recognition," *PRL*, vol. 30, no. 15, pp. 1424–1433, 2009.

[8] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," in *ASFCS, 1984.*, 1984, pp. 338–346.

[9] J. Lladós, E. Valveny, G. Sánchez, and E. Martí, "Symbol recognition: Current advances and perspectives," in *GREC*, 2002, pp. 104–128.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[11] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *IJDAR*, vol. 5, no. 1, pp. 39–46, 2002.

[12] J. Puigcerver, A. Toselli, and E. Vidal, "Word-graph-based handwriting keyword spotting of out-of-vocabulary queries," in *ICPR*, 2014, pp. 2035–2040.

[13] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *CVPR*, 2003, pp. 521–527.

[14] J. M. Romeu, B. Lamiroy, G. Sanchez, and J. Lladós, "Automatic adjacency grammar generation from user drawn sketches," in *ICPR*, vol. 2, 2006, pp. 1026–1029.

[15] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Efficient segmentation-free keyword spotting in historical document collections," *PR*, vol. 48, no. 2, pp. 545–555, 2015.

[16] X. Wang, Y. Tang, S. Masnou, and L. Chen, "A global/local affinity graph for image segmentation," *IEEE TIP*, vol. 24, no. 4, pp. 1399–1411, 2015.

[17] D. Willems, R. Niels, M. van Gerven, and L. Vuurpijl, "Iconic and multi-stroke gesture recognition," *PR*, vol. 42, no. 12, pp. 3303–3312, 2009.

[18] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE TPAMI*, vol. 35, no. 1, pp. 28–38, 2013.